AMENDMENT TO THE CLAIMS

Please amend the presently pending claims as follows:

1. (Currently Amended)   A command processor on a computer system comprising:

a graphical user interface for providing a graphical interface to the computer system; and

a command interpreter, which loads one or more configuration commands into the command processor from at least one of a user specified command configuration script comprising the one or more configuration commands or from a command line in which the one or more configuration commands are entered by the user, interprets the configuration commands and ~~for interpreting commands from a user and for modifying~~ modifies the graphical user interface according to the interpreted configuration commands.

2. (Currently Amended)   The command processor of claim 1 wherein the graphical user interface is modifiable by the user at run time of the graphical user interface through the configuration commands loaded by the command interpreter.

3. (Currently Amended)   The command processor of claim 1 wherein the command interpreter interprets the user configuration commands to produce graphical objects within the graphical user interface.

4. (Currently Amended)   The command process of claim 3 wherein the command interpreter interprets the user configuration commands to assign functionality to the graphical objects.

5. (Original)   The command processor of claim 1 and further comprising:

a suite of integrated circuit design tools, each design tool of the suite having a functionality corresponding to one or more steps in a design flow process of an integrated circuit.

6. (Currently Amended)   The command processor of claim 5 wherein the command processor

loads each design tool into the graphical user interface based on <u>the</u> user <u>configuration</u> commands.

7. (Currently Amended)   The command processor of claim 1 and further comprising:

a graphics engine tool for drawing contents of a database into the graphical user interface based on ~~a user~~ <u>the user configuration commands</u> command.

8. (Currently Amended)   A method of providing a fully customizable graphical user interface comprising:

upon execution of a command processor, loading a top level <u>Tool Command Language</u> <u>(</u>TCL<u>)</u> command into a namespace;

<u>loading one or more TCL commands into the command processor from at least one of a</u> <u>user specified TCL command configuration script comprising the one or more TCL</u> <u>cconfiguration commands or from a command line in which the one or more TCL</u> <u>configuration commands are entered by the user;</u>

building graphical objects according to <u>the</u> TCL <u>configuration</u> commands;

assigning functionality to the built graphical objects according to <u>the</u> TCL <u>configuration</u> commands; and

displaying a user-interactive window containing the graphical objects according to <u>the</u> TCL <u>configuration</u> commands.

9. (Original)   The method of claim 8 and further comprising:

performing functions based on user interactions with the graphical objects according to their assigned functionality.

10. (Original) The method of claim 8 wherein the graphical objects are selected from a group consisting of windows, window panes, buttons, and menus.

11. (Currently Amended)   The method of claim 8 wherein the step of assigning comprises:

creating ~~a TCL~~ the TCL command configuration script corresponding to a circuit design

function; and

assigning the TCL command configuration script to one of the graphical objects.

12. (Original) The method of claim 11 wherein the one of the graphical objects is a button.

13. (Original) The method of claim 11 wherein the one of the graphical objects is an item within a pull-down menu.

14. (Original) The method of claim 8 and further comprising:

changing a look and feel of the graphical user interface during a circuit design process.

15. (Currently Amended)   The method of claim 14 wherein the step of changing comprises:

creating new graphical objects, previously undefined by the command processor, using the

TCL configuration commands; and

assigning functionality to the new graphical objects.

16. (Currently Amended)   The method of claim 14 wherein the step of changing comprises:

loading a new top level TCL command into the namespace, which corresponds to one or

more new TCL configuration commands;

building graphical objects according to the new ~~top-level~~ TCL configuration commands;

assigning functionality to the built graphical objects according to the new TCL

configuration commands; and

displaying the user-interactive window containing the graphical objects according to the

new TCL configuration commands.

17. (Canceled)

18. (Canceled)

19. (Currently Amended) A method of providing a graphical user interface ~~having no hard coded objects~~, the method comprising:

    loading a top level <u>Tool Command Language (TCL)</u> command into a namespace upon execution of a command processor;

    providing a command interpreter for interpreting <u>one or more configuration</u> commands from a user;

    <u>loading the one or more configuration commands into the command processor from at least one of a user specified command configuration script comprising the one or more configuration commands or from a command line in which the one or more configuration commands are entered by the user;</u> and

    assembling a graphical user interface <u>having no hard coded objects</u> based on <u>the</u> interpreted <u>configuration</u> commands from the user;

    wherein all objects within the graphical user interface are user defined <u>through the one or more configuration commands</u>.

20. (Currently Amended) The method of claim 19 and further comprising:

    changing the graphical user interface based on changing <u>configuration</u> commands from the user; and

    displaying a changed graphical user interface during operation based on the changing <u>configuration</u> commands.

21. (Original) The method of claim 19 and further comprising:

    interfacing with a suite of integrated circuit design tools for producing a integrated circuit layout and associated netlist.

22. (Original) The method of claim 21 wherein the step of interfacing comprises:

loading a design tool from the suite of design tools into the graphical user interface based on a user command.

23. (Currently Amended)   The method of claim 22 wherein the <u>one or more</u> user <u>configuration</u> command<u>s</u> ~~is~~ assigned to ~~a graphical object~~ <u>one or more graphical objects</u>.

24. (Currently Amended)   An integrated circuit software design suite comprising:

a command processor having a graphical user interface and a command interpreter for interpreting user commands, the graphical user interface specified entirely by a user <u>through one or more configuration commands loaded into the command processor</u> at run time <u>of the command processor and interpreted by the command interpreter</u>; and

one or more design tools corresponding to processes within an integrated circuit design process;

wherein the one or more design tools operate under control of the command processor and within the graphical user interface.